UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/986,330	11/08/2001	Menahem Lasser	246/100	5581

7590 07/22/2004

DR. MARK FRIEDMAN LTD.
C/O BILL POLKINGHORN
DISCOVERY DISPATCH
9003 FLORIN WAY
UPPER MARLBORO, MD 20772

EXAMINER	
CHU, GABRIEL L	

ART UNIT	PAPER NUMBER
2114	

DATE MAILED: 07/22/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/986,330	LASSER, MENAHEM	
	Examiner	Art Unit	
	Gabriel L. Chu	2114	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 November 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-28* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,6-8,12-20 and 24-28* is/are rejected.

7)☒ Claim(s) *3-5,9-11,13,14 and 21-23* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *4*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Claim Objections

Claim Rejections - 35 USC § 112

1. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

2. Claims 13 and 14 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Referring to claim 13, it is not clear if a command is generated to said device driver. The term "possibly" makes it unclear whether the limitation is to be considered, and, further, if there are any conditions that must be present for at least one command to be generated. For the purpose of examination "possibly generating at least one command" is understood to refer to "generating at least one command".

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1, 2, 6-8, 12, 15-20, and 24-28 rejected under 35 U.S.C. 102(b) as being anticipated by US 5778168 to Fuller. Referring to claim 1, Fuller discloses a method for writing or otherwise changing data in a non-volatile storage device supported by a block device driver so as to provide ruggedized operation, the method comprising the steps

of: a. sending a command to the device driver for defining current data contents of the storage device as a fall-back state in case of failure (From line 44 of column 1, "In a particular embodiment disclosed herein, the OS file system informs the transaction device driver when a file system operation begins and ends."); b. sending a sequence of one or more commands to the device driver, each command potentially changing the data contents of the device (From line 47 of column 1, " During the operation, the file system also informs the transaction driver about important data updates that have occurred since the beginning of the file system operation."); and c. sending a command to the device driver for defining the resulting data contents of the storage device as a new fall-back state in case of failure (From line 44 of column 1, "In a particular embodiment disclosed herein, the OS file system informs the transaction device driver when a file system operation begins and ends.").

Referring to claims 2 and 8, Fuller discloses if a failure occurs after step (a) but before the completion of step (c), the device driver rolls back the effects of all said commands issued in step (b) and returns the storage device to contain said data contents defined as a fall-back state in step (a) (From line 53 of column 1, "Should the system fail while there are outstanding operations, the transaction device driver then ensures that either all of the changes for the operation will appear in the file system or that none of the changes will appear. Consequently, the data written to the disk and encountered at reboot will either be all new data or all old data with no non-deterministic mixing of the two.").

Referring to claims 6, 12, 24, and 27, Fuller discloses ruggedness capability of the device driver can be instructed to be turned on or off (From line 60 of column 1, "Among the benefits of the transaction device driver disclosed herein is that the journaling functionality disclosed may be readily added to any UNIX.RTM. or related OS file system with existing code paths remaining intact with but a few calls to the transaction device driver." Further, from line 65 of column 2, "As described herein with respect to a specific embodiment of the present invention, UFS on-disk format may be retained, no changes are required to add logging to an existing UFS file system and the log can subsequently be removed to return to standard UFS. Moreover, UFS utilities continue to operate as before and file systems do not have to be checked for consistency at boot time.").

Referring to claim 7, Fuller discloses a method for writing or otherwise changing data in a unit-based non-volatile storage device supported by a block device driver so as to provide ruggedized operation, the method comprising the steps of: a. sending a command to the device driver for defining current data contents of the storage device as a fall-back state in case of failure (From line 44 of column 1, "In a particular embodiment disclosed herein, the OS file system informs the transaction device driver when a file system operation begins and ends."); b. sending a sequence of one or more commands to the device driver, each command potentially changing the data contents of the device (From line 47 of column 1, " During the operation, the file system also informs the transaction driver about important data updates that have occurred since the beginning of the file system operation."); and c. sending a command to the device

driver for defining the resulting data contents of the storage device as a new fall-back state in case of failure (From line 44 of column 1, "In a particular embodiment disclosed herein, the OS file system informs the transaction device driver when a file system operation begins and ends.").

Referring to claim 15, Fuller discloses a method for converting an existing non-ruggedized file system on a non-volatile storage device supported by a ruggedized block device driver, into a ruggedized file system, the method comprising the steps of: a. adding, in the beginning of the file system code implementing each file system command which might change data contents of the storage device, new code for sending a command to the device driver for defining the storage device's current data contents as a fall-back state in case of failure; and b. adding, at the end of the file system code implementing each file system command which might change said data contents of the storage device, new code for sending a command to the device driver for defining the storage device's current data contents as a fall-back state in case of failure (From line 44 of column 1, "In a particular embodiment disclosed herein, the OS file system informs the transaction device driver when a file system operation begins and ends." Further, from line 60 of column 1, "Among the benefits of the transaction device driver disclosed herein is that the journaling functionality disclosed may be readily added to any UNIX.RTM. or related OS file system with existing code paths remaining intact with but a few calls to the transaction device driver.").

Referring to claims 16 and 26, Fuller discloses the resulting ruggedized file system is compatible with said existing not ruggedized file system on the same physical

device, such that either a physical device operated under said ruggedized file system can be operated under said existing file system, or a physical device operated under said existing file system can be operated under said ruggedized file system, without loss of data when changing between one file system and the other file system (From line 60 of column 1, "Among the benefits of the transaction device driver disclosed herein is that the journaling functionality disclosed may be readily added to any UNIX.RTM. or related OS file system with existing code paths remaining intact with but a few calls to the transaction device driver." Further, from line 65 of column 2, "As described herein with respect to a specific embodiment of the present invention, UFS on-disk format may be retained, no changes are required to add logging to an existing UFS file system and the log can subsequently be removed to return to standard UFS. Moreover, UFS utilities continue to operate as before and file systems do not have to be checked for consistency at boot time.").

Referring to claim 17, Fuller discloses a method for a software application to write or otherwise change data on a non-volatile storage device, where the storage device is supported by a ruggedized block device driver and a file system, so as to provide ruggedized operation of the application, the method comprising the steps of: a. sending a command to the device driver for defining the storage device's current data contents as a fall-back state in case of failure (From line 44 of column 1, "In a particular embodiment disclosed herein, the OS file system informs the transaction device driver when a file system operation begins and ends."); b. sending a sequence of at least one command to the file system, each said command potentially changing said data

contents of the device (From line 47 of column 1, " During the operation, the file system also informs the transaction driver about important data updates that have occurred since the beginning of the file system operation."); and c. sending a command to the device driver for defining the resulting data contents of the storage device as a new fall-back state in case of failure (From line 44 of column 1, "In a particular embodiment disclosed herein, the OS file system informs the transaction device driver when a file system operation begins and ends.").

Referring to claim 18, Fuller discloses A method for converting an existing non-ruggedized application using a non-volatile storage device supported by a ruggedized block device driver and a file system, into a ruggedized application, the method comprising the steps of: a. adding, before code sending any sequence of commands to the file system which might change the file system's data contents, new code for sending a command to the device driver, which defines current data contents of the storage device as a fall-back state in case of failure; and b. adding, after said code sending any sequence of commands to the file system which might change the file system's data contents, new code for sending a command to the device driver, which defines current data contents of the storage device as a fall-back state in case of failure (From line 44 of column 1, "In a particular embodiment disclosed herein, the OS file system informs the transaction device driver when a file system operation begins and ends." Further, from line 60 of column 1, "Among the benefits of the transaction device driver disclosed herein is that the journaling functionality disclosed may be readily added to any UNIX.RTM. or related OS file system with existing code paths remaining

intact with but a few calls to the transaction device driver.").

Referring to claim 19, Fuller discloses a system providing ruggedized operation of a non-volatile storage device, comprising: a. physical non-volatile storage media; and b. a software block device driver which is ruggedized by itself, independently of a file system or other software application using it (From line 25 of column 1, "Because computer mass storage devices, such as disk drives, cannot guarantee atomicity of data if a system failure should occur during a "write" operation, conventional journaling file systems must use complex transaction mechanisms to compensate. A system failure during a disk write operation usually results in a non-deterministic mix of old and new data having been written to the disk and then subsequently encountered at reboot of the computer system. Consequently, it would be highly desirable if atomicity of data could be guaranteed without the necessity of implementing various complex transaction mechanisms. SUMMARY OF THE INVENTION The transaction device driver technique for a journaling file system of the present invention is of a special utility in ensuring atomicity of write operations to a computer mass storage device in the event of system failure by exporting a transaction interface tailored to the requirements of conventional journaling file systems. In a particular embodiment disclosed herein, the OS file system informs the transaction device driver when a file system operation begins and ends. During the operation, the file system also informs the transaction driver about important data updates that have occurred since the beginning of the file system operation. The transaction device driver herein disclosed logs the updates as the data appears through the normal read/write/strategy entry points into the driver.").

Referring to claim 20, Fuller discloses said physical non-volatile storage media is unit-based media (From line 57 of column 2, "The present invention also improves synchronous write performance by reducing the number of write operations and eliminating disk seek time. Writes are smaller because deltas are recorded in the log rather than rewriting whole file system blocks. Moreover, there are fewer of the blocks because related updates are grouped together into a single write operation. Disk seek time is significantly reduced because writes to the log are sequential.").

Referring to claim 25, Fuller discloses a system providing ruggedized operation of a file system on a non-volatile storage device, comprising the following: a. physical non-volatile storage media; b. a software block device driver for operating said storage media, said device driver being ruggedized by itself, independently of the file system or other software applications using it; and c. a ruggedized file system wherein ruggedness of said file system is achieved by using the ruggedized features of said block device driver (From line 25 of column 1, "Because computer mass storage devices, such as disk drives, cannot guarantee atomicity of data if a system failure should occur during a "write" operation, conventional journaling file systems must use complex transaction mechanisms to compensate. A system failure during a disk write operation usually results in a non-deterministic mix of old and new data having been written to the disk and then subsequently encountered at reboot of the computer system. Consequently, it would be highly desirable if atomicity of data could be guaranteed without the necessity of implementing various complex transaction mechanisms. SUMMARY OF THE INVENTION The transaction device driver technique

for a journaling file system of the present invention is of a special utility in ensuring atomicity of write operations to a computer mass storage device in the event of system failure by exporting a transaction interface tailored to the requirements of conventional journaling file systems. In a particular embodiment disclosed herein, the OS file system informs the transaction device driver when a file system operation begins and ends. During the operation, the file system also informs the transaction driver about important data updates that have occurred since the beginning of the file system operation. The transaction device driver herein disclosed logs the updates as the data appears through the normal read/write/strategy entry points into the driver.").

Referring to claim 28, Fuller discloses a system providing ruggedized operation of a software application on a non-volatile storage device, comprising the following: a. physical non-volatile storage media; b. a software block device driver for operating said storage media, said device driver being ruggedized by itself, independently of the file system or other software applications using it; c. a file system; and d. a software application, such that ruggedness of said application is achieved by using ruggedized features of said block device driver (From line 25 of column 1, "Because computer mass storage devices, such as disk drives, cannot guarantee atomicity of data if a system failure should occur during a "write" operation, conventional journaling file systems must use complex transaction mechanisms to compensate. A system failure during a disk write operation usually results in a non-deterministic mix of old and new data having been written to the disk and then subsequently encountered at reboot of the computer system. Consequently, it would be highly desirable if atomicity of data could be

guaranteed without the necessity of implementing various complex transaction mechanisms. SUMMARY OF THE INVENTION The transaction device driver technique for a journaling file system of the present invention is of a special utility in ensuring atomicity of write operations to a computer mass storage device in the event of system failure by exporting a transaction interface tailored to the requirements of conventional journaling file systems. In a particular embodiment disclosed herein, the OS file system informs the transaction device driver when a file system operation begins and ends. During the operation, the file system also informs the transaction driver about important data updates that have occurred since the beginning of the file system operation. The transaction device driver herein disclosed logs the updates as the data appears through the normal read/write/strategy entry points into the driver.").

Allowable Subject Matter

5. Claims 3-5, 9-11, and 21-23 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. Referring to claims 3, 9, and 21, the prior art does not teach or fairly suggest, in light of their respective parent claims, the device driver identifies data associated with said commands conducted after establishing said fall-back state, by establishing chains of physical blocks associated with the driver's virtual blocks, and storing all new data in said physical blocks, such that said new data is stored in said physical blocks that are not the first blocks in said chains of physical blocks.

Referring to claims 4, 10, and 22, the prior art does not teach or fairly suggest, in light of their respective parent claims, the device driver identifies data associated with said commands conducted after establishing said fall-back state, by associating a ruggedness field with each physical block and detecting changes in the value of said ruggedness field along chains of physical blocks associated with the driver's virtual blocks, such that all new data is in blocks which are positioned after points of said changes.

Referring to claims 5, 11, and 23, the prior art does not teach or fairly suggest, in light of their respective parent claims, the device driver identifies data associated with said commands conducted after establishing said fall-back state, by associating a generation field with each physical block and maintaining a global generation state, such that all new data is in blocks whose generation field equals said global generation value.

6. Claims 13 and 14 are objected to as containing rejected subject matter, but would be allowable if rewritten to overcome the rejected subject matter. Referring to claims 13 and 14, the prior art does not teach or fairly suggest a. optionally examining each command received by said file system, for determining whether said command should be protected from failures in the scope and context of claim 13.

Conclusion

7. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

US 5712971 to Stanfill et al.

US 5802364 to Senator et al.

US 5857204 to Lordi et al.

US 5878428 to Copeland et al.

US 5931954 to Hoshina et al.

US 6282605 to Moore

US 6332200 to Moor et al.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Gabriel L. Chu whose telephone number is (703) 308-7298. The examiner can normally be reached on weekdays between 8:30 AM and 5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Robert W. Beausoliel, Jr. can be reached on (703) 305-9713. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

gc

ROBERT BEAUSOLIEL
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100